IN THE UNITED STATES DISTRICT COURT
THE DISTRICT OF MASSACHUSETTS

NETVIEW TECHNOLOGIES, INC.,	)	
Plaintiff,	)	
v.	)	Case No. 1:09-cv-12072-DPW
MICROSOFT CORPORATION,	)	
Defendant.	)	

PLAINTIFF NETVIEW'S PRELIMINARY CLAIM CONSTRUCTION BRIEF

Respectfully Submitted,

Howard J. Susser (BBO # 636183)
Stephen Y. Chow (BBO # 082990)
Douglas E. Chin (BBO # 671675)
Zachary R. Gates (BBO # 677873)

BURNS & LEVINSON LLP
125 Summer Street
Boston, MA 02110
Tel: 617.345-3000
Fax: 617.345.3299

Attorneys For Plaintiffs

TABLE OF CONTENTS

Plaintiff NetView Technologies, Inc. ("NetView") respectfully submits its claim construction brief pursuant to the Court's Order of May 10, 2010.

I. INTRODUCTION

NetView's asserted U.S. Patent No. 7,251,776, was issued on July 31, 2007 ("the '776 Patent"), based on a July 13, 2001, patent application. It is directed to utilizing in a novel way the conventional, electronic spreadsheet exemplified by that used with the ubiquitous Excel® program marketed by defendant Microsoft Corporation ("Microsoft") for more than a generation. The invention of the '776 Patent is the "parameterization" of a conventional spreadsheet in a "design time" computer environment, with the familiar interactive grid of cells, for use as a program function in a very different "run-time" computer environment, with external input possible only through the parameters.

As explained in the '776 Patent, this method for use of the conventional spreadsheet allows ordinary spreadsheet designers – who typically have business function expertise not including computer programming – to prepare a "parameterized workbook" for use in a run-time environment, where many users in an enterprise may simultaneously and remotely interact with the workbook *simply* by inputting their parameter values, so they can receive customized results (charts, graphics, numerical data). In stark contrast to design-time designer-users, the run-time users cannot modify or even see the formulas and internal workings of the secured spreadsheet model that an enterprise may deem to be proprietary.

In November 2006, with much fanfare, Microsoft launched Office 12, which contained momentous changes to its flagship Excel program to specifically allow spreadsheet designers to do precisely what NetView's '776 Patent confers. Office 12 allows spreadsheet designers to parameterize and publish spreadsheet models for these same run-time usages in a new offering

called "Excel Services" as part of Microsoft's hugely successful SharePoint® business collaboration platform. In countless internal and public pronouncements, and in its worldwide patent applications – filed several years after NetView's – Microsoft touted precisely the same patentable advantages of the NetView claimed inventions asserted here.

NetView construes the disputed claim terms through the eyes of the person having ordinary skill in the art of computer programming who would design the system using the '776 Patent, recognizing key objectives, including that (1) the invention concerns the ordinary electronic spreadsheet; (2) the "parameterization" of the ordinary spreadsheet (workbook) is its preparation for use as a programmatic (computer) function of the parameters in the "run-time" computer environment; and (3) the parameterization is performed at "design time" in a different computer environment of interactivity appropriate for the ordinary spreadsheet model designer without programming skills.

At the same time claiming that the '776 Patent claims are unpatentably abstract in a pending motion for summary judgment, Microsoft avails itself of numerous concrete technical embodiments in the '776 Patent that it impermissibly reads into the claims to avoid liability. Many of these propositions seem more like sleight of hand, and even absurd when viewed by one with knowledge of how computer programs work.

II. BACKGROUND OF THE INVENTION

A. Spreadsheets and Spreadsheet Programs

The technology involved in the'776 Patent was described at pages 4-8 of NetView's opposition (Dkt. # 32) to Microsoft's pending motion for summary judgment for patent ineligibility under 35 U.S.C. § 101 (Dkt. # 27). There, NetView described how the object of Claim 1 of the '776 Patent is the conventional electronic spreadsheet that when opened with the conventional spreadsheet program, such as Microsoft's widely popular Excel, presents the highly

interactive and flexible interface that has been a basis for its popularity since its introduction in the 1970's.[1]

Because the spreadsheet is presented by the spreadsheet program, the term "spreadsheet" is often conflated with the spreadsheet program. Where, as here, two computer (program) environments are involved using the same spreadsheet model, the "spreadsheet" is the information (data and logic specifying computations) that is typically moved or stored as "a file" — such as associated by the ".xls" extension with the Excel 2003 spreadsheet program. When loaded in a computer with the conventional spreadsheet program the spreadsheet presents the familiar interactive grid of cells.[2]

---

1 *See* Exhibit A, '776 Patent, col. 1, ll. 23-53 & 62-65:

> Generally, spreadsheet application programs are used to automate numerical and symbolic calculations for business, financial and scientific uses. Spreadsheet programs are the tools of choice for many business and analysis tasks because they combine a very usable graphical interface with a simple formula language that allows non-programmers, within the limits of the simple formula language, to create computational models. Spreadsheet programs visually present numeric and non-numeric data in a two-dimensional grid for easy assimilation by the reader. Each element of the two-dimensional grid is referred to as a cell. A cell can contain either a data value, or it can contain a formula that calculates a new value based on the values of other cells. Spreadsheet cells that contain formulas are automatically recalculated when there are changes to the other cells that the formula depends upon. This mechanism allows a spreadsheet user to perform what-if scenarios only by modifying cell values and viewing or saving the effects of the changes. Individual two-dimensional spreadsheets can be organized into a larger entity known as a notebook or workbook. The terms spreadsheet or worksheet will be used interchangeably herein, as will the terms notebook or workbook. When worksheets are grouped together to form a workbook, the workbook and all of its worksheets are stored together as a single file (i.e. the workbook becomes the unit of storage and transfer when moving data between the program 's memory space and disk storage). Formulas stored in worksheet cells can reference other cells that are in worksheets in the same workbook and/or cells that are in worksheets in a different workbook In existing spreadsheet programs, data values can be stored in cells either by user input (directly or through a user-input formula) or by the user associating the cells with an external data source, such as a query to a database.

2 Even in this ordinary use with a conventional spreadsheet program, multiple "copies" of this spreadsheet are used. "Loading" or "opening" of an .xls file with the Excel program generates at least a temporary copy or "image" or "instance" of the file that is loaded into the active memory space of the Excel program. Other features may call for additional copies.

Just as we have all experienced the horror of failing to store our edits, enterprises with many users of spreadsheets and other documents that are stored and opened locally at their desktops may make local changes to a supposedly authoritative version and incorrectly store the changes, resulting in different versions. This is particularly troublesome with a complex spreadsheet model. As the '776 Patent notes (Exh. A, col. 6, ll. 45-61):

> One example of a business problem is the use of spreadsheets to manage sales commission programs. In a typical situation, each salesperson's commission plan may be based on several variables, such as sales quota goals or particular commission rates to be paid on certain sales. Each salesperson 's plan may also vary based on their seniority or the kind of territory they cover. In current practice, compensation specialists often model the commission plans using spreadsheets. Ideally, each salesperson will have a separate spreadsheet customized to their situation. Managers will also have their own tailored commission plans, modeled as a spreadsheet, and these will often depend on the results of the people reporting to them. The result is a computational model that consists of a large web of interdependent spreadsheets, which can number in the thousands for a large sales organization.

In a traditional use of a spreadsheet, the compensation specialist/spreadsheet designer may just email a copy of the spreadsheet model to all the sales representatives, who open the spreadsheet in their local computer program environments, edit and add values to cells, pull appropriate data from data sources, and allow the spreadsheet program to automatically recompute their output. These users are "*in" the spreadsheet*, learning all the information in the model, including formulas and databases the enterprise considers proprietary. These users may modify an originally authoritative or "true" version of the model and may circulate or use outdated versions erroneously, as multiple versions of the proprietary model float around unsecured in the enterprise. This lack of uniformity and control presents a critical challenge to continuity of an enterprise's operations.

The '776 Patent provides the following novel solution: the enterprise designer creates a single version of the spreadsheet model in the ordinary way and prepares it in that design environment for a prescribed, limited use in a different run-time environment. Enterprise users

wishing to utilize the model to get a customized result can access it on-demand, but they never receive "the" spreadsheet, nor do they even see the proprietary model. They can do this remotely over the Internet without having a copy of the spreadsheet program. The designer "parameterizes" the model so that users can submit their particular "*external parameter*" values to the model, which is "called" and executed as a computer function on the enterprise server. The "parameterized workbook" is called and loaded as an instance that receives the user's values, recomputes the new spreadsheet values based on their data (which may involve extracting data from other databases), and returns to the caller a customized output. In accordance with its operation as a function, the output is not a copy of the spreadsheet for interactive execution in a conventional spreadsheet program environment, but is typically a web page (HTML) document that contains limited display results previously designated by the designer to be visible to run-time users.[3]

---

3 *See* '776 Patent, Exh. A (col. 7, ll. 25-33): "The ability to access these spreadsheet models on demand allows self-service applications to be created for information consumers. For example, using a self-service web site, a salesperson can access their current commission calculations or a chief financial officer can view and download an up to date projection of the commission expenses for the current quarter." As stated in the file history in response to the Examiner's inquiry regarding tangible results of the claimed invention (Exhibit B, at NETVIEW000041-000042):

> As described in the examples in the specification, a typical use of the present invention is to automate repetitive and manual spreadsheet-based business processes. The present invention allows a spreadsheet user to avoid having to manually modify a spreadsheet many times and instead allows the user to set up one parameterized workbook that can automatically make all of the required calculations. As a concrete example, consider a person who needs to calculate and distribute ten commission reports, each for a separate business unit, on a weekly basis, where each commission report is a spreadsheet that is similar in form, but with varying contents. Rather than having to prepare ten separate spreadsheets every week, the present invention allows this person to create one parameterized workbook at design time that implements the ten commission reports. The parameterized workbook would have two parameters: the name of the business unit and the current (or requested) date. Once the parameterized workbook is created, the present invention can, at run-time, automatically compute and display the ten commission reports for each business unit, each and every week. In a typical embodiment, each report is computed and displayed on demand in response to a web form where someone using a web browser can request a report at any time for a particular business unit and a particular week.

To the computer software developers who are the persons of ordinary skill in the art to which the '776 Patent is addressed (the people who would build the *system* required to practice the methods claimed), the patent is directed to a *functionalized spreadsheet* that is called as a computer function that "maps" an input parameter value to an output value. *See* Exh. A, '776 Patent, col. 13, ll. 17-18 ("In other words, the parameterized workbook 105a is a function mapping a set of inputs 110 to a set of outputs 120").

In the design-time environment of the claimed method, where visibility of and ability to manipulate cells are crucial, the spreadsheet is displayed to the spreadsheet model designer in its ordinarily mode: a grid of hundreds of cells, each of which accepts input of values (such as numbers and strings of characters) and formulas (specifying mathematical and other operations on "contents" of cells as well as data in other specialized "locations" in the spreadsheet) and which automatically updates the resultant values of formulas.

In the run-time environment, the same model information in the "parameterized workbook" may be called and loaded with the run-time computer program as a function in multiple instances for simultaneous executions – with external inputs initiated by each caller only through a few parameters and with designated results of the spreadsheet-model-specified computation returned to the caller.[4]

---

4 Unlike the spreadsheet at design time, which is open to input to any cell by the user, this run-time use of the spreadsheet in essence encapsulates the spreadsheet (or more precisely its "instance" in a spreadsheet-computing program environment), building a "black box" around the model, and allowing input only through the defined parameter "window" through which parameters values are input from an "outside" thereby defined. Thus, the same information or spreadsheet, used in the different run-time computer environment does not present the ordinary interactive grid, but a "black box" computer function with parameter "windows" through which input may be made from outside.

B. NetView Patent Prosecution History

The file history of the '776 Patent contains procedural and substantive prosecution events. Substantively, the Patent Office raised two main rejections. First, the PTO initially objected to the applicant's proposed subject matter pursuant to 35 U.S.C. § 101 (the same statute relied upon by Microsoft's in its pending motion for summary judgment). Second, the PTO initially rejected the claim based on prior art. Both were overcome.

In its Section 101 rejection, the PTO relied on an earlier standard of "useful, tangible, concrete" results which it deemed was met with the amendments to Claim 1, adding "run-time" utilization of the spreadsheet, leaving intact the original claim elements regarding the parameterization specifically to be performed in a distinct "design time." Microsoft's current § 101 challenge has been fully briefed -- with the design-time configured machine and the run-time configured machine as particular machines overcoming any *Bilski* challenge (Dkt. #32).

As to prior art, the PTO chiefly raised an IBM patent issued in 2003 ("Jamshidi", U.S. Patent 6,631,497, attached hereto as Exhibit C), entitled "Binding Data From Data Source To Cells In A Spreadsheet." *See* Exh. B, at NETVIEW000104. This rejection was overcome by NetView, which filed a clarifying amendment explaining that Jamshidi disclosed the use of a type of "parameter" for initiated a call or query from the spreadsheet instance to a data source – that is, the Jamshidi "parameter" is "of the data source. In contrast, the "external" parameter of Claim 1 of the '776 Patent is "of" the "spreadsheet as a whole," and because it is the input to the parameterized spreadsheet from outside the spreadsheet at run-time, it is "external" to the spreadsheet at that time. *See* Exh. B, at NETVIEW000044-000046. NetView's proposed constructions are consistent with this prosecution history.

III. PATENT LAW

A. Claim Construction Principles

"[T]he construction of a patent, including terms of art within its claim, is exclusively within the province of the court." *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 372 (1996). It is a "bedrock principle" that "the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed.Cir.2005) (en banc) (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed.Cir.2004)). There is a "heavy presumption" that a claim term should be given its ordinary meaning. *See CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002). "[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question *at the time of the invention, i.e.*, as of the effective filing date of the patent application." *Phillips* at 1313 (emphasis added); *Multiform Desiccants, Inc. v. Medzam, Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998) ("It is the person of ordinary skill in the field of invention through whose eyes the claims are construed."). Thus, when the ordinary meaning of claim language as understood by a person of skill in the art is readily apparent to a lay judge, claim construction "involves little more than the application of the widely accepted meaning of commonly understood words." *Id.* at 1314. "In such circumstances, general purpose dictionaries may be helpful," provided "the dictionary definition does not contradict any definition found in or ascertained by a reading of the patent documents." *Id.* at 1314, 1322-23 (quoting *Vitronics*, 90 F.3d at 1584 n. 6).

However, if this meaning is not readily apparent, the court should review "the intrinsic evidence of record, *i.e.*, the patent itself, including the claims, the specification and, if in evidence, the prosecution history." *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582

(Fed. Cir. 1996). First, the court should "look to the words of the claims themselves, both asserted and non-asserted, to define the scope of the patented invention." *Id.*

Next, the court should "review the specification to determine whether the inventor has used any terms in a manner inconsistent with their ordinary meaning." *Vitronics,* 90 F.3d at 1582. Because the specification is highly relevant to the claim construction analysis, it is considered to be "the single best guide to the meaning of a disputed term." *Phillips,* 415 F.3d at 1315 (quoting *Vitronics,* 90 F.3d at 1582).

Nonetheless, "[w]hen consulting the specification to clarify the meaning of claim terms, courts must take care not to import limitations into the claims from the specification." *Abbott Labs. v. Sandoz, Inc.,* 566 F.3d 1282, 1288 (Fed.Cir.2009) (en banc). The Federal Circuit has noted that "although the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments." *Phillips,* at 1323. Instead, the focus must remain on the claim language, because "claims may embrace different subject matter than is illustrated in the specific embodiments in the specification." *Id.* (citation omitted). In other words: "claims, not specification embodiments, define the scope of patent protection. The patentee is entitled to the full scope of his claims, and we will not limit him to his preferred embodiment or import a limitation from the specification into the claims." *Kara Tech. Inc. v. Stamps.com Inc.,* 582 F.3d 1341, 1347 (Fed. Cir. 2009).

"When the specification describes a single embodiment to enable the invention, [a] court will not limit broader claim language to that single application unless the patentee has demonstrated a clear intention to limit the claim scope using words or expressions of manifest exclusion or restriction." *Abbott,* 566 F.3d at 1288 (internal quotation marks and citation omitted). *See also Northrop Grumman Corp. v. Intel Corp.,* 325 F.3d 1346, 1355 (Fed.Cir.2003) ("Absent a clear disclaimer of particular subject matter, the fact that the inventor may have

anticipated that the invention would be used in a particular way does not mean that the scope of the invention is limited to that context."). Likewise, "[t]he fact that a patent asserts that an invention achieves several objectives does not require that each of the claims be construed as limited to structures that are capable of achieving all of the objectives." *Phillips*, 415 F.3d at 1327 (quoting *Liebel-Flarsheim v. Medrad, Inc.*, 358 F.3d 898, 908 (Fed. Cir. 2004)).

The final form of intrinsic evidence is the prosecution history, which consists of the complete record of the proceedings before the PTO and includes the prior art cited during the examination of the patent. *Autogiro Co. of Am. v. United States*, 384 F.2d 391, 399 (U.S. Ct. Cl. 1967). "Even when prior art is not cited in the written description or the prosecution history, it may assist in ascertaining the meaning of a term to a person skilled in the art. When prior art that sheds light on the meaning of a term is cited by the patentee, it can have particular value as a guide to the proper construction of the term, because it may indicate not only the meaning of the term to persons skilled in the art, but also that the patentee intended to adopt that meaning." *Arthur A. Collins, Inc. v. Northern Telecom Ltd.*, 216 F.3d 1042 (Fed. Cir. 2000) (citing *Vitronics*, 90 F.3d at 1584).

If the meaning of a claim term remains ambiguous after the intrinsic evidence is consulted, the court may "rely on extrinsic evidence, which 'consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises.'" *Phillips*, 415 F.3d at 1317 (quoting *Markman*, 52 F.3d at 980). However, the Federal Circuit has instructed that extrinsic evidence is "less significant than the intrinsic record in determining 'the legally operative meaning of claim language.' " *Id.* at 1317 (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed.Cir.2004)).

B. Other Principles of Law

Normally, validity issues would not be appropriate at the claim construction stage. However, by seeking to define certain claim terms of the '776 Patent as referring to "mathematical functions," Microsoft has forced NetView to preliminarily revisit Microsoft's motion for summary judgment of patent invalidity for lack of patentable subject matter under 35 USC § 101. That motion has been fully briefed (Dkt. #32), and remains pending -- though Microsoft clings to a fully debunked argument that the '776 Patent claims an unpatentable mathematical algorithm, when no mathematical formula is even recited, much less impermissibly preempted. A recent decision of the Federal Circuit reversing a trial court's invalidation of patents under this provision further compels denial of Microsoft's motion. *Research Corp. Tech. v. Microsoft Corp.*, No. 2010-1037, 2010 U.S. App. LEXIS 24984 (Fed. Cir. Dec. 8, 2010) ("*RTC*").[5]

As in *RTC*, "[t]he invention [here] presents functional and palpable applications in the field of computer technology." *Id.* at *19. The Federal Circuit in *RTC* notes that "inventions with specific applications or improvements to technologies in the marketplace are not likely to be so abstract that they override the statutory language and framework of the Patent Act." *Id.* Here, the '776 Patent is directed to a specific application of the conventional spreadsheet technology in the marketplace, claiming the use of a conventional spreadsheet acted upon with two particular machines (computers distinctly configured for "design time" and "run-time"). This clearly is not "abstract" nor preemptive of an "abstract idea."

5 Microsoft was represented by the same lead counsel in that appeal, making largely the same patent-ineligibility arguments as in this action. The *RTC* court rejected those arguments on largely the same basis as argued by NetView here.

Microsoft also has made references to its "indefiniteness" position in pleadings; however, it has done so in an *indefinite* and incomplete way, and NetView has objected to Microsoft's lack of proper pleading. Therefore, NetView respectfully reserves the right to address any argument raised by Microsoft that may require the Court to address the "indefiniteness" of claim terms. *Atmel Corp. v. Information Storage Devices, Inc.*, 198 F.3d 1374, 1378 (Fed. Cir. 1999) ("A determination of claim indefiniteness is a legal conclusion that is drawn from the court's performance of its duty as the construer of patent claims.").

IV. CLAIM TERMS FOR CONSTRUCTION

A. Introduction

Claim 1 and dependent claim 10 are presently asserted by NetView. Each claim is directed to a method that is used on a system containing the two computer environments corresponding to: (1) "design time" and (2) "run-time."[6]

1. A method for utilizing a spreadsheet, the method comprising: defining a parameter external to the spreadsheet; associating the parameter with the spreadsheet at design time to define a parameterized workbook, wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter; receiving a value for the parameter at run-time; computing cell values in the spreadsheet that are dependent, directly or indirectly, on parameter; and rendering an output based on the computed cell values.	10. The method of claim 1, further comprising: defining a formula within the spreadsheet at design time using the parameter.

6 NetView has expressly reserved the right to assert Claim 18, which is a dependent claim directed to a "parameterized workbook information database" based on incomplete discovery from Microsoft. Microsoft objects to this reservation of rights.

1. The Computer Environments and the Person Having Ordinary Skill in the Art

The '776 Patent contemplates three classes of individuals relevant to the claimed invention. At the top level is the software developer to whom the claimed invention is directed; the person of ordinary skill in the art.[7] This is the person who builds/implements a system to practice the method of Claim 1.[8] This person by definition will write software code to create a system (build or configure the two particular machines) that can implement the method.

The next level is the spreadsheet model designer/creator. This is the design-time user who creates and stores the parameterized workbook at design time; the one in an enterprise who practices the design-time steps of the claimed method. He/she is a *non-programmer*. The specification undeniably explains that the method is practiced on a system that allows non-programmer design-time "creators" to take a conventional spreadsheet, "parameterize" it, and make it accessible as a function for users at run time. For example, as stated in Exhibit A, '776 Patent:

> Spreadsheet programs are the tools of choice for many business and analysis tasks because they combine a very usable graphical interface with a simple formula language that allows non-programmers, within the limits of the simple formula language, to create computational models. Col. 1, ll. 25-30.
>
> Some programs publish application programming interfaces (APIs) that allow computer programs to be written to manipulate the spreadsheets or to extend the user interface of the program. Because programming skills are required to use these application programming interfaces, they are not used by most spreadsheet users. Col. 2, ll. 17-23.

---

7 *See* Exh. A, '776 Patent, col. 1, ll. 16-19 ("TECHNICAL FIELD. This invention relates generally to computer-based systems and methods for data processing, and more particularly relates to systems and methods for manipulating data, for example, spreadsheet application programs.").

8 The '776 Patent discussed the "system" and the method, and the original patent application contained system claims. However, during prosecution the patent applicant, acting *pro se*, assented to withdrawal of all system claims in an examiner's amendment upon allowance of the method claim.

> Among other advantages, the invention described above allows non-programmers greater flexibility, including allowing the application of spreadsheets to certain kinds of business problems that are not tractable with conventional spreadsheet programs. Col. 6, ll. 31-35.
>
> The systems include spreadsheet modeling mechanisms that work in concert to allow non-programmers to model classes of problems that are intractable using prior art spreadsheet programs. Col. 6, ll. 42-45.
>
> Introducing parameterized spreadsheets 105 as a formal modeling mechanism provides an interface suitable for non-programmers to create reusable spreadsheet-based computational building blocks and it provides programmers with increased flexibility and power. Col. 12, ll. 34-39.

The third class is the run-time user, who is not expected to be a spreadsheet designer, much less a computer programmer. He/she is able to "utilize" the parameterized workbook model to get a customized output while at a location remote from the enterprise that may not even be a personal computer. Exhibit A, '776 Patent, col. 11, ll. 62-65 ("A parameterized workbook 105 calculation may be initiated in response to a user request from a web browser or another user interface device including cell phones, personal digital assistants, etc.").

As discussed below, these manifest objectives present in the '776 Patent are borne out in NetView's proper construction of the distinct and limited *design-time* and *run-time* computer environments.

2. "Spreadsheet" And "The Spreadsheet"

The "spreadsheet" is the object the particular use of which is at the heart of the claimed invention. It is clear to the person of skill in the art that this is the conventional spreadsheet used with the conventional spreadsheet program, such as Excel. Thus, NetView proposes a definition of "spreadsheet" that explains to the trier of fact what a conventional electronic spreadsheet is and does:

> Data embodying a model which used with a conventional spreadsheet program on a computer presents an interactive graphical user interface showing one or more two-dimensional grids of cells through which a user may enter values and formulas, where each cell has an associated value or optionally a formula that specifies the cell value,

> where the formulas may refer to the values in other cells, and where the formulas may be automatically recomputed upon changes to values of cells upon which they depend.

This definition summarizes the salient features of the spreadsheet used with conventional spreadsheet programs as explained in the specification (column 1, quoted above), and makes clear that it is the spreadsheet information specifying the spreadsheet model that is referred to throughout the claims (see text at note 2 *supra*). This model is acted upon by at least two particular machines, the design-time machine that presents a conventionally interactive spreadsheet, and the run-time machine in which the spreadsheet ("parameterized workbook") is executed, but not seen (a.k.a. "headlessly"). (Another machine contemplated is that used by a run-time user, for example a personal digital assistant, to initiate input of parameter values and to receive an output.)

Until yesterday, Microsoft proposed for "spreadsheet:" "A single two-dimensional grid of cells forming a logical unit of calculation. Each cell can contain a formula for calculating a data value. Each cell not containing a formula can instead contain a data value." While reciting a grid so that it is superficially similar to NetView's construction, this definition is indefinite and unduly broad: It is not clear how a "grid of cells" forms "a logical unit of calculation" and Microsoft apparently would have it extend to non-conventional abstract "logical unit[s] of calculation" proposed as substitutes for actual ordinary spreadsheets (unsurprisingly similar to Microsoft's asserted prior art), when '776 Patent could hardly be more clearly directed to what the Federal Circuit in *RTC* called a "improvement of technology [already] in the marketplace." Thus, it is clear from the '776 Patent description that the "spreadsheet" referred to in all the claims of the '776 Patent is confined to the one used with conventional spreadsheet programs, such as Excel, and does not extend so far as to include "new paradigms" replacing conventional spreadsheets *in toto*, that Microsoft may contend (in reference to prior art it has raised) still

might form "a logical unit of calculation."[9] Yesterday, Microsoft amended its proposal to require that the formula it recites must calculate a "data value from another data value." No such requirement appears in the claims or specifications, and with a day's notice, NetView can only surmise that the proposed limitation might somehow excuse Microsoft from infringement.

Claim 1 introduces the term "a spreadsheet" in the preamble: "A method for utilizing a spreadsheet, the method comprising" and the claim goes on with steps that make use of "the spreadsheet" at design time and run-time. On this basis, Microsoft seeks to limit the claim to a single physical copy of the spreadsheet, which must be parameterized and used at run-time (Microsoft: "the spreadsheet" means "the same spreadsheet that is referenced as 'the spreadsheet' elsewhere in the claim, not a different copy of it.").[10]

While linguistically there is a single "the spreadsheet" stated, the "spreadsheet" here is the data/logic (constituting a model) used to present an interactive interface at design time and to direct computations at run-time: it is the same data even if it is recorded at different places. Microsoft's contention that there can only be one copy of the spreadsheet in the claims is absurd as: (1) it is well understood in the art that a certain type of "copy" is always made of such a file to run any kind of program without affecting the characterization that there is one "the file" (see

---

9 Microsoft's Excel is specifically mentioned in the original provisional patent application incorporated by reference into the '776 Patent, *see* Exhibit D, p. 2, and an Excel spreadsheet is clearly depicted in the upper left corner of Figure 4 of the '776 Patent. The reason why Microsoft seeks to broaden this main claim term is to ensnare any number of prior art references that are not conventional spreadsheets but provide some – but not all—the advantages of the '776 Patent, including the use of the familiar conventional spreadsheet at design time (by an ordinary spreadsheet designer).

10 Microsoft has sought to avoid infringement on this basis in its contentions: "The accused operations identified by NetView in its infringement contentions do not perform claims' steps *vis-a-vis* 'the spreadsheet,' to the extent that 'the spreadsheet' allegedly refers in this claim to the same (unspecified) allegedly physical instantiation of that abstract function, rather than merely to the same abstract function with different allegedly physical "copies" or instantiations thereof." Dkt. 39-1, pp. 1-2.

note 2 *supra*); and (2) the patent is directed to locking down a single *model* that is the parameterized workbook that may be utilized by multiple run-time users simultaneously.

As to the first point, when a Word or Excel document or file is "opened" with its application program, it is the temporary copy (or "instance") that is "loaded" into active memory. There may be other "copies" used for memory management, all of which would still be referred to as "the file".

As to the second point, the patent clearly teaches that multiple "copies" of "the" parameterized workbook (embodying "the spreadsheet") can be executed simultaneously:

> These prior art spreadsheet programs do not allow multiple copies of the same workbook to be loaded into memory simultaneously and they do not allow multiple what-if scenarios to be calculated simultaneously. Exh. A, '776 Patent, col. 1, ll. 57-61.
>
> When a parameterized workbook 105 calculation is performed in the context of a specific set of parameters 110, this can be referred to as a workbook instantiation. As illustrated with path 135, a model 100 can simultaneously instantiate a parameterized workbook 105a multiple times with different values for the parameters 110. Exh. A, '776 Patent, col. 12, ll. 40-45.

This recognizes that, while the *original* parameterized workbook is persistently stored, *it* can be and is expected to be used in simultaneous executions at run time, requiring multiple execution copies or instances (in addition to the ones required for a single execution).[11] This is the nature of having a centralized, "locked-down" parameterized workbook called by multiple enterprise users to be executed as a computer function with their individual instances.

Microsoft's proposed definitional requirement that the spreadsheet receiving parameter value at run time not be a "different copy" of the spreadsheet incorporated into the parameterized

---

11 *See also, e.g.*, Exh. A, '776 Patent, col. 20, ll. 57-61: "The equation in the second portion 415 uses the parameters 110 'person' and 'period' as part of the formula. When the system 300 instantiates the workbook, the system 300 binds the value of the parameters person and period sent with the call to generate *that instance*." (Emphasis added.)

workbook at design time makes no sense for a single execution of a single copy, much less for its use as a computer function called by multiple users at run time.

B. The "Design Time" Environment

Both parties contend that the term "design time" in the claim is directed to the part of the claimed process where a creator creates the "parameterized workbook." Microsoft would stop there (i.e., "when the parameterized workbook is defined."), redundantly marking the step of "associating the parameter…to define a parameterized workbook." However, the patent repeatedly makes clear that it is directed at allowing a *non-programmer* spreadsheet designer to avoid having a software programmer or other IT professional program special interfaces to link together each spreadsheet so that it can be used as a function at run-time.[12] Claim 1 must be understood to require that the "design time" environment is one familiar to the ordinary spreadsheet model designer and does not require such *ad hoc* code programming every time a spreadsheet is to be parameterized for use as a function at run-time. Thus, NetView proposed the following construction to reflect the clear objective of the invention::

> the time when a spreadsheet model is created and parameterized; here, through direct user access to spreadsheet cells in a conventional spreadsheet user interface in a computer environment further configured to allow the user to functionalize the spreadsheet without resorting to code programming.

NetView's construction recognizes that a "designer" is both creating the spreadsheet model and converting it into a parameterized workbook without traditional *ad hoc* code programming. The '776 Patent describes a system that allows a non-programmer to *functionalize* a workbook –

---

12 *See* Exh. A., '776 Patent, at col. 2, ll. 12-14 & 20-23 ("Some prior art spreadsheet programs support various facilities for programmatic control over the spreadsheets to automate spreadsheet tasks. . . . Because programming skills are required to use these application programming interfaces, they are not used by most spreadsheet users."), and col. 6, ll. 31-35 ("Among other advantages, the invention described above allows non-programmers greater flexibility, including allowing the application of spreadsheets to certain kinds of business problems that are not tractable with conventional spreadsheet programs.").

turning a conventional workbook into a type of computer function that can be run by callers, who provide parameter values and receive output from a spreadsheet model. It is not about writing a new computer program for each spreadsheet. The method claims of the patent utilize that system; in essence, a non-programmer practices the design time step.

1. "Defining a Parameter External to the Spreadsheet"

The design-time steps of the claimed process generally include two subparts: *defining* the external parameter, and *associating* it with the spreadsheet to lock it down as a parameterized workbook. For purposes of Claim 1 of the '776 Patent, the existence of a spreadsheet is generally presupposed. In the order written, the claim first requires "defining a parameter external to the spreadsheet." The "external parameter" is the "window" (*see* note 4, *supra*), through which the user at run-time may input a value to the parameterized workbook, in contrast to the traditional way such as directly interacting with, for example, a spreadsheet opened in Excel by typing that value "into" one of the spreadsheet's cells visible to the user on screen. Thus, NetView construes the *defining* term as follows:

> formally declaring a named input variable to a spreadsheet by which values will be passed from outside the spreadsheet when the spreadsheet is executed at run-time as a function that maps an input value to an output value

The "parameter" of this invention is functionally and existentially "external" to the spreadsheet that is used at run-time. Stated differently, at run-time the spreadsheet is encapsulated as a "black box" and no longer accepts input directly into every cell as in its ordinary use with a conventional spreadsheet program such as Excel. Instead, it receives input only through the external parameter "window" that allows an outside caller to input values into the run-time encapsulated spreadsheet-and-calculation engine. The "externality" is that a

parameter value is passed from outside the spreadsheet-as-a-function to inside. Thus, a "parameter" is referred to as an "external input" in the'776 Patent, Exh. A, col. 8, ll. 48-49.[13]

Yet, Microsoft's proposed construction appears to focus not on the function or purpose of the external parameter in the utilization of the claimed method ("1. a method for utilizing a spreadsheet…"), but apparently on the *physical existence* <u>*at all times*</u> of something separate from (a single copy of, in its construction) "the spreadsheet" even when the spreadsheet is not being run as a function. Microsoft asserts the following two-part construction:

> "*defining a parameter*": Establishing the meaning of a parameter, including its identity and type, but not a specific value of the parameter;
>
> "*a parameter external to the spreadsheet*": A parameter existing and defined independently of the spreadsheet and its contents, such that the parameter's definition does not refer to the spreadsheet and eliminating the entire spreadsheet does not affect the parameter."

The second construction is especially opaque and to the extent understood, requires that the "external" in "external parameter" means the parameter must "exist" and be "defined independently" of the spreadsheet. This is a misdirected focus on where information regarding the parameter is stored when the spreadsheet is dormant that ignores the parameter's actual function and use at run time. The function of the parameter of claim 1 is to "parameterize" the

---

13 As the patent file history makes clear, the "externality" of the parameter is with reference to **<u>values being supplied externally</u>**:

> The invention described in [the prior art] Jamshidi is fundamentally different for the present invention because the present invention discloses a method for creating a parameterized workbook, not a parameterized data binding. In a parameterized workbook, the parameters are associated with and apply to a spreadsheet as a whole. Any formula in the spreadsheet can reference the parameters. ***The parameter values are supplied externally to the computation of the spreadsheet as a whole***, whereas with the parameterized data source bindings in Jamshidi, the parameter values are specified by formulas within the spreadsheet itself.

Exh. B, at NETVIEW000045 (emphasis added).

spreadsheet so that, **at run time**, a parameter value can be provided into the spreadsheet – the parameter cannot exist solely "independently" of the spreadsheet in any meaningful sense.

First, based on the objective of the "external parameter" as described in the intrinsic record, as discussed throughout this brief, there is no basis to construe this term to ignore its function and objective in favor of alleged storage locations or file formats. At best, Microsoft's position may be boiled down to an improper attempt to read into the claims its characterization of a preferred embodiment.[14] Microsoft's proposed physical location limitation of where a parameter "exists" relative to the "spreadsheet" requires a "parameterized workbook" *must be* comprised of three separate and distinct "modules" (see discussion i*nfra*).

Second, there is no basis to saddle the construction of this defining step with an indefinite negative limitation regarding "elimination" (physically? logically?) of the spreadsheet where there is no deletion test stated in the claims or in the specification. The further limitation – "the parameter's definition does not refer to the spreadsheet" is another red herring. The Patent is about parameterizing a spreadsheet, yet, according to Microsoft the parameter's definition *must not* "refer to" the spreadsheet for which it is created? [NetView respectfully reserves the right to address in its responsive brief this issue first proposed by Microsoft on January 6, 2011 and others proposed January 9, 2011 – days before this brief was due, though the parties first exchanged proposed claim constructions on November 19, 2010, per Court Order.]

Regarding the Microsoft's "defining" construction, there is no requirement in the patent that a parameter must have an identity (name) and type as Microsoft contends. The specification

---

14 Microsoft looks to read this extraneous language into the claims because Microsoft contends that it stores information *about* its external parameter within the collection of zipped files it presently calls a spreadsheet file (collected in a format known as .xlsx).

is clear that parameters "may" be "typed" and that it is an option that the creator "can" associate a "type" with each workbook parameter.[15]

2. "Associating the Parameter with the Spreadsheet To Define a Parameterized Workbook"

The next part of the process in the order presented in the claim is the association step:[16] "associating the parameter with the spreadsheet at design time to define a parameterized workbook." The following table shows the disputes:

	NetView Construction	Microsoft Construction
parameterized workbook	A persistently stored spreadsheet based model that acts as a programmatic function at run-time to map an external parameter input value to an output value, that may be called by multiple users or programs for multiple and simultaneous executions without changing the persistent model.	A mathematical function for mapping defined inputs to defined outputs, and having three separate but associated logical modules: a parameter module defining a parameter, a workbook module defining a spreadsheet, and a results (outputs) module defining a named result having a type. Prior to run-time, its parameter has no specific value.

15 *See* Exh. A, '776 Patent: "Workbook parameters may be typed, and the type may limit the supplied values" (Col. 8, ll. 3-4); "The creator can also associate type information with each workbook parameter 110 and result 120. The type information constrains the set of legal data values the system can use for a parameter 110 when instantiating a workbook and the set of legal data values that the system can return as an output result 120 from a workbook 105 instantiation." (Col. 9, ll. 14-19). In addition, the doctrine of claim differentiation supports this conclusion. Claim 8 is dependent upon claim 1, and recites: "The method of claim 1, further comprising: associating a type with the parameter at design time, wherein the type defines a range of values that are allowable to be received for the parameter at run-time." By reading in the requirement of a "type" definition into claim 1, Microsoft would render claim 8 redundant.

16 The sequence in which such steps are written is not a requirement unless as a matter of logic or grammar, they must be performed in the order written, or the specification directly or implicitly requires such a narrow construction. *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323 (Fed. Cir. 2001). NetView submits that the order of the defining and associating steps is not critical provided the parameterized workbook is defined at design time for use at run-time.

associating the parameter with the spreadsheet at design time to define a parameterized workbook	Persistently storing the spreadsheet based model and relationship between it and the external parameter that allows execution instances of the persistent spreadsheet model to be called as a programmatic function at run-time by multiple users or programs for multiple and simultaneous executions in a computer program environment, where input to the function is provided only through the external parameter.	At design time, associating the existing, defined external parameter with the spreadsheet. Until this step is performed, the parameter is not associated with the spreadsheet.

Repeating its already debunked summary judgment contention, Microsoft states that a "parameterized workbook" is a "mathematical function." As shown in NetView's opposition to summary judgment (Dkt. # 32), there is no recitation of any mathematical algorithm representing any "natural law," much less any impermissible preemption of such a mathematical algorithm. Rather, each individual parameterized workbook acts as a *programmatic* function, that is, a type of subroutine in computer programming (a procedure called repeatedly in a persistent form) mapping an externally input parameter value to an output value (the patent does not use the term "mathematical" at all).[17] Thus, the first part of NetView's construction is the one true to the specification: "A persistently stored spreadsheet based model that acts as a programmatic function at run-time to map an external parameter input value to an output value."

---

17 NetView in its summary judgment briefing set forth the dictionary differences between mathematical and programmatic functions. (Dkt. #32, at pp. 3-4). There is nothing in the '776 Patent that states that the parameterized workbook is a mathematical function rather than the computer program function that computer programmer recognizes that the '776 Patent parameter workbook actually becomes in the run-time environment. *See* Exh. A, '776 Patent, col. 13, ll. 17-25: "[A] parameterized workbook 105a (FIG. 1) creates a set of outputs 120 (FIG. 1) based on a set of input parameters 110 (FIG. 1). In other words, the parameterized workbook 105a is a function mapping a set of inputs 110 to a set of outputs 120. The outputs 120 may be visual, such as a chart or a report rendered in some format (e.g. GIF, HTML) intended for display or printing. The outputs 120 may also be data values represented in some format (e.g. XML) intended for further processing."

Claim 1 does not recite "modules" at all and does not imply the necessity of such organization. Microsoft nonetheless seeks to read in a "three-module" limitation to the term "parameterized workbook" ("having three separate but associated logical modules: a parameter module defining a parameter, a workbook module defining a spreadsheet, and a results (outputs) module defining a named result having a type.") – presumably to distinguish the structure of the .xlsx collection of files (see note 14 *supra*) it uses with the alleged infringing process. Microsoft's reading of an "existing" parameter and that "prior to run-time . . . has no specific value" is cut from whole cloth, found neither in the claims nor in the specification, apparently to distinguish Microsoft's particular implementation of the '776 process.

In NetView's '776 Patent, the concept of a module is conducive to high-level description, but there is no *requirement* that information (about parameters) be stored in particular places, whether logical modules, hardware or software. Rather, the patent specification disclaims any rigid limitation to specific storage schemes:

> The modules throughout the specification can be implemented as a software program (e.g., a set and/or a sub-set of processor instructions and the like) and/or a hardware device (e.g., ASIC, FPGA, processor, memory, storage device and the like). … Each of the servers and modules 305, 310 and 315 can be any computing device capable of providing the services requested by the other servers and modules or by the client node 320. Particularly, this includes generating and processing parameterized workbooks as described herein.

Exh. A, '776 Patent, col 8, ll. 63-67 and col. 15, ll. 48-52. The '776 Patent broadly teaches that an *association* between the external parameter and the workbook is not limited to an association between separate parameter and workbook modules (Exh. A, '776 Patent, col. 10, line 60 – col. 11, line 1) (emphasis added):

> This example file names the parameters 110 associated with the parameterized workbook 105 Rep ProductLineCommissionCalculations "person" and "period" and defines them as "Person" and "Year" types, respectively. Person and Year have definitions in a data dictionary that describes the object model of the application. *In this example, the*

> *association 125 of the parameter 110 with the workbook 115 is made in the workbook itself in addition to being described in the file*.

The bottom line is the '776 specification does not limit the patent invention to requiring use of modules nor to any particular type of module. *See* Exh. A., '776 Patent, col. 11, ll. 38-41 ("In other embodiments, the workbook module 115 can include any logical unit of spreadsheet computation, comprising one or more spreadsheets and associated formulas, as the unit of parameterization."). Depending on the conventional spreadsheet used, there may be different "logical units." Indeed, the '776 specification describes more than the three modules, including in addition to the three noted by Microsoft, an instantiator module (col. 3, line 34), a selection module (col. 4, line 36 and col. 13, line 7), and a calculations module (col. 4, line 41). Yet, Microsoft does not attempt to read in all of these.

Microsoft trips over itself to manufacture an indefiniteness defense by presupposing that "modules" must be read into the '776 Patent claim 1, but it cannot figure out whether it should be a three-module or two-module "entity" (Dkt. # 39, at pp. 27-28):

> On the one hand, the patent's written description repeatedly states that 'parameterized workbook' is a three-module entity, including a separate but associated results module.... But, elsewhere, the applicants implied that the 'parameterized workbook' refers to a two module entity, comprising the workbook module and parameter module, and not the results module. ... Thus, although the 'three-module entity' definition appears to be the proper one, the intrinsic evidence on this is inconsistent, leaving the person of skill in the art to guess as to the true meaning.

Rather than evidence of "indefiniteness" of a claim that doesn't even mention "module", this is an admission that (to Microsoft) the intrinsic record teaches that these are *embodiments* – that a parameterized workbook *may* include two or three "modules" so conceptualized.

Microsoft's construction also ignores explicit language in the '776 Patent specification describing that the parameterized workbook is accessible as a function in a run-time environment

by *multiple users*, who can simultaneously utilize the function, while maintaining the original parameterized workbook as the persistent model, see *supra*, at pp. 16-17.

3. "To Define A Parameterized Workbook, Wherein The Parameter Applies To The Spreadsheet As A Whole Thereby Allowing Any Formula In The Spreadsheet To Reference The Parameter"

NetView submits that the quoted element is understood to have its ordinary meaning in the context of the whole claim and the objective of the parameterized workbook, which is that a run-time user initiates the parameterized workbook as a function by inputting external parameter values, so that an instance of the spreadsheet can be *instantiated* (data acquired, formulas computed, etc.), and a new customized result can be outputted. This subject element, "the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter," is merely a recitation to this function. See Exhibit A, '776 Patent, col. 11, ll. 1-7 (a workbook called "Rep_ProductLineCommissionCalculations" with external parameters "person," and "period"):

> [T]he spreadsheet Rep_ProductLineCommissionCalculations contains formulas that use the parameters person and period in them. When the system instantiates the parameterized workbook 105, the system binds specific values for these parameters to the instantiation of the Rep_ProductLineCommissionCalculations spreadsheet.

Later in the specification it refers to Figure 4, where the formula referencing the parameters by name is illustrated:

> FIG. 4 illustrates an embodiment of a screen shot 400 of a specific example of parameterized workbook 105 (FIG. 1) that the system 300 (FIG. 3) can generate and process in accordance with the invention. The screenshot 400 includes a first portion 405 indicating the name of the workbook containing the associated spreadsheet formulas and a second portion 415 indicating an exemplary formula used for producing descriptive text based on the workbook parameters 110 and a third portion 420 showing the resulting descriptive text displayed in a cell. ….
>
> The equation in the second portion 415 uses the parameters 110 "person" and "period" as part of the formula. When the system 300 instantiates the workbook, the system 300 binds the value of the parameters person and period sent with the call to generate that

> instance. The result of the formula is a descriptive text string based on the workbook parameter values, shown in a cell in the third portion 420.

Exh. A, '776 Patent, col. 20, ll. 33-42 & 56-62. Plainly, this is all that is required by this claim element and NetView construed the element to mean "an ordinary formula in the spreadsheet model is capable of referring directly or indirectly to the named parameter."

Microsoft dishonestly seeks to narrowly limit this term:

	NetView construction	Microsoft construction
…wherein the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter	an ordinary formula in the spreadsheet model is capable of referring directly or indirectly to the named parameter.	"**the parameter applies to the spreadsheet as a whole, thereby allowing any formula in the spreadsheet to reference the parameter**": All cells of the spreadsheet and all formulas in the spreadsheet can reference the parameter. "**reference the parameter**": Bind directly to the external parameter without referring to a cell in the spreadsheet.

First, Microsoft requires that "all cells of the spreadsheet can reference the parameter," Second, Microsoft reads "any formula in the spreadsheet to reference the parameter" to contain the negative limitation "[but] without referring to a cell in the spreadsheet."

Neither of these limitations are in any way suggested much less compelled by the claim language or the specification. One can only surmise that this is just a creation of a loophole because Microsoft apparently passes parameter values through designated "precursor" cells of its spreadsheets as one of many possible implementations (this one technically characterized as "indirection"). However, under the issued language of Claim 1, it is only required that cells with formulas may refer to the parameter – the use of the ordinary spreadsheet as a function where each formula in the spreadsheet may accept the parameter value. There is no basis to read in Microsoft's specially tailored limitations based on repeated admonitions by the Federal Circuit.

Dependent claim 10 recites a further limitation closely related to the foregoing: "10. The method of claim 1, further comprising: defining a formula within the spreadsheet at design time using the parameter." The salient point regarding this claim 10 is in its differentiation from claim 1, as to which the parties appear to agree. Claim 10 requires that a formula actually reference a parameter by name. Claim 1 thus only requires that a parameterized workbook is created that ***is capable of*** using a formula that directly references the parameter by name, not that it actually does.

C. The "Run-time" Environment

Fundamentally, NetView's '776 Patent requires that the initiation (calling) of a parameterized workbook and passing of external parameter values at run-time is not through manual intervention by entering data directly into cells of a spreadsheet in a spreadsheet program environment:

> Two objectives of the present invention are to simplify the creation and maintenance of large spreadsheet-based computation models and to allow those models to be computed *without manual intervention*.

Exh. A, '776 Patent, col. 16, ll. 4-8 (emphasis added). Rather, a parameterized workbook is utilized from outside that typical spreadsheet program environment. That is, "run-time" is (Netview's proposed construction):

> a time during which the functionalized spreadsheet is executed as a function; here, in a computer environment that allows multiple users or programs to make multiple and simultaneous calls and executions where input is provided only through the external parameter from outside that environment by a user or a running program (caller) not using a conventional spreadsheet program that allows for direct user modification of cells.

The parameterized workbook is *locked-down* at run-time in that it is only executed through specifying the parameter values, which are passed from the caller to the calculation server where

an instance of the parameterized workbook model is instantiated based on the parameter values, so that a unique rendered output is returned back to the user.

The '776 Patent states:

> An end user, a formula appearing in a workbook cell, and/or a program using an API can initiate a parameterized workbook 105 calculation either interactively or non-interactively. Other computer systems communicating over a network can also initiate a parameterized workbook 105 calculation. These other systems may use established communication protocols such as, for example, CORBA (Common Object Request Broker, from OMG (Object Management Group)), RMI (Java Remote Method Invocation) or SOAP (Simple Object Access Protocol, from Microsoft). A parameterized workbook 105 calculation may be initiated in response to a user request from a web browser or another user interface device include cell phones, personal digital assistants, etc.

Exh. A, '776 Patent, col. 11, ll. 52-65. Thus, NetView submits that this run-time environment requires that "input is provided only through the external parameter from outside that environment by a user or a running program (caller) not using a conventional spreadsheet program that allows for direct user modification of cells."[18]

Microsoft contends that "run-time" means "when the spreadsheet is used for computation." While this contention is technically accurate (run-time is the time something runs), it is overly simplistic for this patent – and would apply to ordinary use of the spreadsheet

---

18 The reference to initiation of a parameterized workbook calculation by "a formula appearing in a workbook cell" is limited to the patent's presently unasserted claimed inventions directed to nested or cascading parameterized workbooks, where a first parameterized workbook is initiated by a user (or non-spreadsheet program) and the instance of that first parameterized workbook calls a second parameterized workbook to run. In such as case, the second parameterized workbook is initiated from within a spreadsheet program. However, manifestly, the first parameterized workbook must still be initiated from outside the spreadsheet program environment. See Exh. A, '776 Patent, col. 12, ll. 42-51 ("As illustrated with path 135, a model 100 can simultaneously instantiate a parameterized workbook 105a multiple times with different values for the parameters 110. This allows a parameterized workbook 105 calculation to depend on one or more calculations from subsidiary parameterized workbooks (e.g., 105a, 105b...105n). Any subset of the subsidiary instantiations, and even the referencing workbook instantiation itself, can be instantiations of the same parameterized workbook 105.").

in an ordinary spreadsheet environment.[19] The run time of the '776 Patent is for running the parameterized workbook as a computer function – with input initiated only through the parameter *window* between the user and the workbook.

1. "Receiving A Value For The Parameter At Run-time"

The run-time environment requires that the "receiving a value for the parameter" step of the process initiate at its origin from a user or program outside the normal spreadsheet program environment. Thus, NetView contends that "receiving a value for the parameter at run-time" means "input of an external parameter value to an execution instance of the functionalized spreadsheet as loaded in its execution environment from outside that environment by a user or a running program (caller) not using a conventional spreadsheet program that allows for direct user modification of cells."

Microsoft contends the element means three things: "Receiving a value for the external parameter. This step happens at run-time not at design time. This step cannot be varying the value of a spreadsheet cell." Taking these three contentions one at a time, the first part may be correct, but it is insufficient. The person of ordinary skill in the art understands that the claim is not directed to direct user interface access to the spreadsheet, so the origin of the receiving step is understood as outside that domain.

The second part ("this step happens at run-time not at design time.") is generally consistent with the claimed invention in the sense that the environments are different and in the normal use of the invention design time and run-time are temporally distinct. However, the main

19 As in the other instances in which it ignores the express invention of the '776 Patent, Microsoft obviously attempts here to ensnare any type of environment, including direct user access to spreadsheet cells in the prior art asserted by Microsoft.

point is that design-time and run-time are *environmentally* distinct, and there does not appear to be any reason to read in a temporal limitation based on the intrinsic record.[20]

Microsoft's third part ("this step cannot be varying the value of a spreadsheet cell") is a negative limitation that has no basis in the claims or the description, and is yet another blatant attempt to read out Microsoft's alleged infringing process. Thus, as mentioned above, Microsoft apparently utilizes designated cells to pass parameter values to the spreadsheet being used as a function. Apparently to pass the parameter values, Microsoft varies the value of a spreadsheet shell. Thus, Microsoft seeks to add the tailored negative limitation that NetView's claim 1 "receiving a value for the parameter" ***must not*** be done that way to practice claim 1.

This is improper as there are no words of manifest exclusion or unambiguous disavowal in the patent or file history that prevent such a precursor step at design time or run-time. The point is that the external parameter value comes from the external source, and that such value is delivered to the parameterized workbook in such a way that it can be recomputed, and a formula can use that parameter by name.

2. "Rendering An Output Based On The Computed Cell Values"

The run-time use claimed in claim 1 of the spreadsheet is the execution of the spreadsheet as a function (that is, the spreadsheet opened or loaded in its run-time calculation/rendering

---

20 The Patent File History actually states that design time and run-time can roughly coincide when, for example, a designer is testing a run of the parameterized workbook. *See* Exh. B, at NETVIEW000040-000041 ("Claim 1 is also amended to be more specific with respect to the timing of the steps, making a distinction between 'design time' and 'run-time' as is common in the art. The amended claim 1 states that the definition and association of the parameter happens at 'design time,' while receiving the parameter value and computing dependent cells within the spreadsheet happens at 'run-time.' As is common in the art, design time and run-time may be nearly coincident in certain embodiments of the present invention, for example to provide the user with immediate feedback at design time. In most embodiments, however, it is expected that design time and run-time will involve distinct periods and typically different users. In the present invention, the designer of a parameterized workbook is typically creating a spreadsheet which will be computed and displayed many times by other run-time users."). Thus, the claimed method must be able to have disparate design and run-time, but an infringing system does not avoid infringement by having them coincide.

program), mapping an input value to an output *value*. Thus, NetView submits that "rendering an output" is "returning to the caller results computed from the spreadsheet (as visual displays and/or data values, but not as a spreadsheet) based on the input external parameter value." This is consistent with the point of the patent that the run-time user is calling the spreadsheet-as-a-function, not using an ordinary spreadsheet program or *directly* accessing the spreadsheet itself:

> The output 120 of a parameterized spreadsheet 105 calculation can take many forms. The software performing the parameterized spreadsheet 105 calculation can format a worksheet or a region on a worksheet for display on some output device. The display format can vary depending on the output device, and can include standardized output formats such as HTML (Hypertext Markup Language, the main document format recognized by web browsers), WML (Wireless Markup Language, similar to HTML but targeted at wireless devices such as cell phones), or XML (extensible Markup Language, used for business to business (or system to system) communication), in addition to device specific formats. Alternatively, the output 120 of a parameterized spreadsheet 105 calculation can be a set of data values, suitable for use in further data processing. Various output formats can be used for representing these sets of data values 120, including document formats such as HTML or XML or formats based on data communication protocols such as CORBA or RMI. Parameterized spreadsheets 105 allow greater control over the course of the computation, including the ability to select different sets of external data upon which to operate.

Exh. A, '776 Patent, col. 12, ll. 5-26. Noticeably missing from the foregoing list of outputs is a spreadsheet file for use with a spreadsheet program.[21] In other words, what is returned to the caller of the spreadsheet-as-a-function is output value to which the parameter input is mapped, not the spreadsheet itself. If the spreadsheet itself were the rendered output of its run-time use as

---

21 The patent specification provides an example of a typical "output" as follows:

> For example, a particular workbook 105a might have two parameters 110. The first parameter 110 is typed as the person object, further constrained to have the job title of telesales representative. The second parameter 110 is typed as a time period object, further constrained to be a calendar month. The workbook 105a might define several outputs 120. One output 120 might be a bar chart indicating sales performance of the sales person for a particular month (the output type might be specified as a GIF image). Another output might be a list of daily sales performance metrics (such as the number of calls or average call duration or product revenue generated). The type of this second output might be an array of numeric values (one for each day of the month).

Exh. A, '776 Patent, col. 10, ll. 8-22.

a function, this would be outside the "black box" environment of run time, contrary to the base distinction between the ordinary use of the spreadsheet and its run-time use in the '776 Patent.

Until yesterday, Microsoft contended that "rendering an output based on computed cell values" means "[b]ased on the two or more computed cell values, providing outside the parameterized workbook a value for a named result (output) of the parameterized workbook." On Sunday morning, Microsoft deleted the "outside the parameterized workbook" from this construction. This fundamental change in its construction – with consequences that NetView is deprived of having the prescribed time to consider – is a compelling reason to reject Microsoft's constructions.

V. Conclusion

For the reasons stated herein, NetView respectfully asks that its constructions be adopted.

Dated: January 10, 2011

/s/ Howard J. Susser

Howard J. Susser (BBO # 636183)
hsusser@burnslev.com
Jeffrey R. Martin (BBO # 322520)
jmartin@burnslev.com
Stephen Y. Chow (BBO # 082990)
schow@burnslev.com
Merton E. Thompson (BBO # 637056)
mthompson@burnslev.com
Paul T. Muniz (BBO # 564786)
pmuniz@burnslev.com
Laura L. Carroll (BBO # 076180)
lcarroll@burnslev.com
Douglas E. Chin (BBO # 671675)
dchin@burnslev.com
Zachary R. Gates (BBO # 677873)
zgates@burnslev.com

BURNS & LEVINSON LLP
125 Summer Street
Boston, MA 02110
Telephone: 617-345-3000
Attorneys for Plaintiff

CERTIFICATE OF SERVICE

I, Howard J. Susser, hereby certify that on January 10, 2011, a copy of PLAINTIFF NETVIEW'S PRELIMINARY CLAIM CONSTRUCTION BRIEF was served on counsel of record for all parties via email through the CM/ECF system.

/s/ Howard J. Susser
Howard J. Susser (BBO # 636183)